Refine Search

Search Results -

Terms	Documents
L5 or L6	7

Database:
- US Pre-Grant Publication Full-Text Database
- **US Patents Full-Text Database**
- US OCR Full-Text Database
- EPO Abstracts Database
- JPO Abstracts Database
- Derwent World Patents Index
- IBM Technical Disclosure Bulletins

Search: L7

Refine Search

Recall Text | Clear | Interrupt

Search History

DATE: Tuesday, April 13, 2004 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=YES; OP=ADJ		
L7	l5 or l6	7	L7
L6	5,201,040.pn.	1	L6
L5	l3 and L4	6	L5
L4	(6516372 or 6075938 or 5590301 or 5446841 or 5301324 or 4709325).pn.	6	L4
L3	l1 same L2	16	L3
L2	multi-processors or multiprocessor	13008	L2
L1	independent near2 ((os) or (operating adj system))	1531	L1

END OF SEARCH HISTORY

Freeform Search

Database: US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Term: l1 and L4

Display: 15 Documents in Display Format: - Starting with Number 1

Generate: ○ Hit List ◉ Hit Count ○ Side by Side ○ Image

Search Clear Interrupt

Search History

DATE: Tuesday, April 13, 2004 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ			
L5	l1 and L4	38	L5
L4	l2 with simultan$6	9602	L4
L3	l1 and L2	509	L3
L2	(multiple or several or plurality) with ((os) or (operating adj system))	313489	L2
L1	directly with (coupl$5 or connect$5 or access$5) with memory with ((os) or (operating adj system))	1176	L1

END OF SEARCH HISTORY

Refine Search

Search Results -

Terms	Documents
L3 and L4	6

Database:
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L5

Refine Search

Recall Text | Clear | Interrupt

Search History

DATE: Tuesday, April 13, 2004 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=YES; OP=ADJ		
L5	l3 and L4	6	L5
L4	(6516372 or 6075938 or 5590301 or 5446841 or 5301324 or 4709325).pn.	6	L4
L3	l1 same L2	16	L3
L2	multi-processors or multiprocessor	13008	L2
L1	independent near2 ((os) or (operating adj system))	1531	L1

END OF SEARCH HISTORY

L5: Entry 1 of 6 File: USPT Feb 4, 2003

DOCUMENT-IDENTIFIER: US 6516372 B1
TITLE: Partitioning a distributed shared memory multiprocessor computer to facilitate selective hardware maintenance

Abstract Text (1):
A distributed shared memory multiprocessor computer system is provided, which has a number of processors and is divided into partitions. Each partition has within it one or more of the processors, and may also have memory or cache and other related hardware. Although each partition works together and communicates with other partitions to share computational load, the partitions each are independently operable and execute an independent copy of the operating system. The partitions comprise additional features to enable removal of a partition from the operating computer system, and to enable insertion of hardware into the operating computer system.

Brief Summary Text (14):
A distributed shared memory multiprocessor computer system is provided, which has a number of processors and is divided into partitions. Each partition has within it one or more of the processors, and may also have memory or cache and other related hardware. Although each partition works together and communicates with other partitions to share computational load, the partitions each are independently operable and execute an independent copy of the operating system. The partitions comprise additional features as described herein, to enable removal of a partition from the operating computer system, and to enable insertion of hardware into the operating computer system.

CLAIMS:

1. A distributed shared memory multiprocessor computer system, comprising: a plurality of processors; a plurality of partitions, each of which comprises one or more of the plurality of processors, executes an operating system independent of other partitions, and is independently removable from and insertable into the computer system while the computer system remains in operation; a system interconnection network, linking the plurality of partitions; a control module, operable to disable hardware before removal from the operating computer system and to enable hardware after insertion into the operating computer system; and a reset fence that prevents partition reset signals from propagating outside the partition.

7. A distributed shared memory multiprocessor computer system, comprising: a plurality of processors; a plurality of partitions, each of which comprises one or more of the plurality of processors, executes an operating system independent of other partitions, and is independently removable from and insertable into the computer system while the computer system remains in operation, wherein each partition further comprises one or more c-bricks, each c-brick comprising one or more processors, and each partition further comprises one or more routers, such that each router is connected to one or more c-bricks and to at least one other router; a system interconnection network, linking the plurality of partitions; a control module, operable to disable hardware before removal from the operating computer system and to enable hardware after insertion into the operating computer system.

8. A distributed shared memory multiprocessor computer system, comprising: a plurality of processors; a plurality of partitions, each of which comprises one or more of the plurality of processors, executes an operating system independent of other partitions, and is independently removable from and insertable into the computer system while the

computer system remains in operation; a system interconnection network, linking the plurality of partitions; and a control module operable to disable hardware before removal from the operating computer system and to enable hardware after insertion into the operating computer system, wherein the control module is further operable to tristate connections between the partition and the computer system such that the computer system is logically detached from the partition.

9. A distributed shared memory multiprocessor computer system, comprising: a plurality of processors; a plurality of partitions, each of which comprises one or more of the plurality of processors, executes an operating system independent of other partitions, and is independently removable from and insertable into the computer system while the computer system remains in operation; a system interconnection network, linking the plurality of partitions; and a control module, operable to disable hardware before removal from the operating computer system and to enable hardware after insertion into the operating computer system, wherein the control module is further operable to handle errors generated as a result of removal of hardware.

11. A distributed shared memory multiprocessor computer system, comprising: a plurality of processors; a plurality of partitions, each of which comprises one or more of the plurality of processors, executes an operating system independent of other partitions, and is independently removable from and insertable into the computer system while the computer system remains in operation; a system interconnection network, linking the plurality of partitions; a control module operable to disable hardware before removal from the operating computer system and to enable hardware after insertion into the operating computer system; and a cache, wherein the cache is not coherent with cache in other partitions, so that removal of one cache from an operating computer system will not result in the computer system crashing.

L5: Entry 2 of 6 File: USPT Jun 13, 2000

DOCUMENT-IDENTIFIER: US 6075938 A
TITLE: Virtual machine monitors for scalable multiprocessors

Brief Summary Text (6):
The second approach to dealing with the system software challenges of scalable shared-memory multiprocessors is to statically partition the machine and run multiple, independent operating systems that use distributed system protocols to export a partial single system image to the users. An example of this approach is the Sun Enterprise10000 machine that handles software scalability and hardware reliability by allowing users to hard partition the machine into independent failure units each running a copy of the Solaris operating system. Users still benefit from the tight coupling of the machine, but cannot dynamically adapt the partitioning to the load of the different units. This approach favors low implementation cost and compatibility over innovation. Digital's announced Galaxies operating system, a multi-kernel version of VMS, also partitions the machine relatively statically like the Sun machine, with the additional support for segment drivers that allow applications to share

Drawing Description Text (2):
FIG. 1 is a schematic diagram illustrating the architecture of a computer system according to the invention. Disco, a virtual machine monitor, is a software layer between a multiprocessor hardware layer and multiple virtual machines that run independent operating systems and application programs.

Detailed Description Text (57):
Using this mechanism, multiple virtual machines accessing a shared disk end up sharing machine memory, The copy-on-write semantics means that the virtual machine is unaware of the sharing with the exception that disk requests can finish nearly instantly. Consider an environment running multiple virtual machin es for scalability purposes. All the virtual machines can share the same root disk containing the kernel and application programs. The code and other read-only data stored on the disk will be DMA-ed into memory by the first virtual machine that accesses it. Subsequent requests will simply map the page specified to the DMA engine with out transferring any data. The result is shown in FIG. 4 where all virtual machines share these read-only pages. Effectively we get the memory sharing patterns expected of a single shared memory multiprocessor operating system even though the system runs multiple independent operating systems.

L5: Entry 3 of 6 File: USPT Dec 31, 1996

DOCUMENT-IDENTIFIER: US 5590301 A
TITLE: Address transformation in a cluster computer system

Detailed Description Text (10):
The 30-bit register 15 in the primary cache of the CPU 11 can specify one gigabyte of address locations which is the family limit of the exemplary CPUs. There is value in reserving a portion of the addressable space in main memory 8 to the private use of the CPUs on each multiprocessor board for, among other reasons, permitting each multiprocessor board to operate under an independent operating system. (These independent operating systems may be the same or different operating systems.) On the other hand, there is a need for establishing direct communication among the clusters and CPUs operating under the different operating systems in order that they can function with the full power of a cluster system architecture. One way in which this feature can be achieved is by providing shared memory space in main memory 8.

L5: Entry 4 of 6 File: USPT Aug 29, 1995

DOCUMENT-IDENTIFIER: US 5446841 A
TITLE: Multi-processor system having shared memory for storing the communication information used in communicating between processors

Brief Summary Text (4):
On the other hand, for example to an information processing system in which a plurality of processors are operated by independent operating systems, respectively, and are loosely coupled to one another through a common memory (shared memory), is described in Japanese patent un-examined publication No. JP-A-64-78361 (corresponding to copending U.S. patent application Ser. No. 07/209,073 filed Jun. 20, 1988), U.S. Pat. No. 5,201,040. In JP-A-64-78361, it is described that an instruction equal to a signal processor instruction, used for communication between the processors, is used between the processors by the loosely coupled multi-processor system through the shared memory to perform the communication. However, it is not described that a processor number and a shared memory number are dynamically changed to perform the communication. Further, in JP-A-64-78361, it is not also described that when a processor of a virtual machine (hereinafter, referred to as simply "a virtual processor" when applicable) simulated on a real machine performs the communication with another 10 virtual processor through the shared memory, that communication is performed without reading out the storage data of the shared memory in order to recognize a virtual processor number.

Generate Collection Print

L5: Entry 5 of 6 File: USPT Apr 5, 1994

DOCUMENT-IDENTIFIER: US 5301324 A
TITLE: Method and apparatus for dynamic work reassignment among asymmetric, coupled processors

Detailed Description Text (15):
FIG. 1 shows the system environment within which the present invention operates. FIG. 1 shows a tightly coupled multiprocessor system consisting of four processors (CPUs), processor W, 100, processor X, 101, processor Y, 102, processor Z, 103, sharing a common memory, 130. Each of the processors, 100-103, contain features, 110, 111, 112, 113. Processor W, 100 contains features C and E, 110. Processor X, 101, contains features A and D, 111. Processor Y, 102, contains features B, C, and D, 112. Processor Z, 103, contains features A and B, 113. Each of the processors are currently executing work units. The work units are, WU4, 120, on processor W, 100, WU3, 121 on processor X, 101, WU1, 122, on Processor Y, 102, and WU2, 123, on Processor Z, 103. Shown as residing in the common memory, 130, are key data for assigning work units to processors, 131, 132, and 133. These data are the Work Unit Table, 131, the Ready Queue, 132, and the Processor Feature table, 133. In the preferred embodiment the multiprocessor computing system as shown is executing with a hypervisor, 140, such as IBM's Processor Resource/System Manager, PR/SM, facility in Logical Partitioning, LPAR, mode. The Assignment means, 141, and the Promotion mechanism, 142, are, in the preferred embodiment, a program within the hypervisor. In this example the work units being assigned are actually operating systems which are independent of each other. The invention works equally as well, in an alternative embodiment, in a multiprocessor computing system environment running under control of an operating system, such as IBM's MVS/ESA. In this case the Assignment means and the Promotion mechanism are within the operating system and the work units are tasks, address spaces or jobs. The processors use conventional communication paths for signalling each other.

L5: Entry 6 of 6 File: USPT Nov 24, 1987

DOCUMENT-IDENTIFIER: US 4709325 A
TITLE: Loosely coupled multiprocessor system capable of transferring a control signal set by the use of a common memory

Brief Summary Text (3):
A conventional loosely coupled multiprocessor system of the type described is disclosed by James A. Katzman in "A Fault-Tolerant Computing System" published on January, 1979 (first revision), by Tandem Computers Inc., Calif. The conventional loosely coupled multiprocessor system comprises a plurality of processor units having independent operating systems and a plurality of peripheral devices used by the processor units in common. At any rate, each of the processor units can individually carry out a processing operation by the use of a set of control signals.

Detailed Description Text (2):
Referring to FIG. 1, a loosely coupled multiprocessor system according to a first embodiment of this invention comprises first and second processor units 11 and 12 which individually have independent operating systems in the manner known in the art. Each of the first and the second processor units 11 and 12 is operable in accordance with a predetermined program which is similar for the processor units 11 and 12. The program of each processor unit 11 and 12 is executed by using a set of control signals variable with time to process a job imposed on each processor unit 11 or 12. Therefore, the control signal sets of the first and the second processor units 11 and 12 are different from each other and will be referred to as first and second control signal sets, respectively. The first and the second processor units 11 and 12 are coupled to each other through a processor interface line 14.